## AMENDMENTS TO THE SPECIFICATION

**Please replace paragraph [0008] with the following amended paragraph:**

[0008]      In another embodiment, method of forming a rule object can comprise the steps of reading rules from at least one among a library and a textual ~~textural~~ datafile and reading rules from at least one among a dynamic rule component library and a textual ~~textural~~ rule component datafile, and constructing component objects from at least one among the library and the textual ~~textural~~ datafile and from at least one among the dynamic rule object library and the textual ~~textural~~ rule component datafile.  The method can further include the step of matching component objects having a same identifier and revision to form the rule object.  The method can also include the step of binding the matching component objects that have a most recent version at runtime to form the rule object.

**Please replace paragraph [0026] with the following amended paragraph:**

[0026]      Referring to FIG. 3, a flow chart illustrates a method 100 of forming a rule object.  The method 100 can include the steps of reading rules from at least one among a library at step 102 and a textual ~~textural~~ datafile at step 108 and reading rules from at least one among a dynamic rule component library at step 104 and a textual ~~textural~~ rule component datafile at step 110.  In other words, rules can be read from libraries already existing at step 102 and dynamic rule component libraries can also be read in as "plug-ins" at runtime at step 104.  Likewise, rules can be read from textual ~~textural~~ datafiles already in existence at step 108 and textual ~~textural~~ rule component datafiles can be read in as plug-ins at runtime at step 110.  Component objects can be constructed from at least one among the library and the textual ~~textural~~ datafile and from at least one among the dynamic rule object library and the textual ~~textural~~ rule component datafile at steps 106 and 112 accordingly.  The method 100 further matches component objects having a same identifier and revision or version to form the rule object at step 114.  The rule object formed is unique.  The step of matching can comprise the step of binding the matching component objects that have a most recent version at runtime to form the rule object.  The rules that are

constructed can then be stored in a database at step 116.


**Please replace paragraph [0031] with the following amended paragraph:**

[0031]      This invention can be embodied in other forms without departing from the spirit or essential attributes thereof.  Notably, the concepts discussed herein can be applied to other rule-based operations in design software.  For example, systems for performing circuit optimization or circuit transformation triggered by certain circuit topologies can be coded into similar categories such as Application Criteria, Condition, and Action.  Again, these other rule-based operations can be expressed or implemented either using textual ~~textural~~ datafiles or program object code.  Accordingly, reference should be made to the following claims, rather than to the foregoing specification, as indicating the scope of the invention.